Towards Automatic 3D Reconstruction of Pitched Roofs in Monocular Satellite/Aerial Images

Zachary Blair and Parvaneh Saeedi

School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada
{zblair, psaeedi}@sfu.ca

Abstract

This paper presents a novel method for estimating the three-dimensional shape of pitched roofs in monocular satellite/aerial images by utilizing the acquisition geometry (both sun and camera). This method consists of four steps: rooftop extraction, texture and vent removal, identification of planar sections, and 3D roof model generation. During the fourth step, 3D reconstruction candidates are partially rendered, and iteratively compared against the original image to evaluate a fitness function. Experimental results illustrate that the proposed framework can accurately derive 3D building rooftop models. Comparison of the reconstructed rooftop models against ground truth is very promising, indicating an average error within 5% in the slopes of reconstructed models.

1. Introduction

3D roof reconstruction from a single image is a compelling prospect for a variety of purposes, including 3D building reconstruction and modeling, 3D map generation, and urban planning. For instance, insurance companies currently employ assessors to travel to insured properties to estimate building properties including rooftop models and slopes of each piece. The volume, variability and complexity of digital geospatial images mandate fully-automated 3D rooftop modeling and reconstruction capabilities.

Previous approaches to 3D roof reconstruction have typically relied on specialized imaging technologies, such as binocular imaging, or LiDAR (Light Detection And Ranging) [1]. This requirement typically makes these approaches more expensive, because they require specialized equipment.

This paper proposes a novel framework for 3D roof reconstruction from a single image, and builds upon previous work for determining building boundaries automatically [2]. The novelty in this approach is that it requires only a single image for its reconstruction, and it has been implemented using a cross-platform framework to leverage GPU hardware.

1.1. Previous Work

Algorithms that recover a 3D shape from a single 2D image based on shading are said to be “shape-from-shading” (SFS) algorithms, and although SFS is one of the classic problems in computer vision, there is yet to be a single algorithm that performs well in all situations [3]. SFS algorithms have been applied to a variety of problems, from estimating the shape of microscopic objects, to estimating the shape of geographical features [4].

Many previous SFS approaches assume a Lambertian surface, meaning that light reflected from the surface is reflected in all directions equally. Therefore, the pixel intensity, $E(x, y)$ at any point is given by the equation,

$$E(x, y) = \frac{\cos(\sigma) + p \cos(\tau) \sin(\sigma) + q \sin(\tau) \sin(\sigma)}{\sqrt{1 + p^2 + q^2}}$$

(1)

where $\tau$ is the tilt of the illuminant, $\sigma$ is the slant of the illuminant, and $p$ and $q$ define the surface orientation in terms of the partial derivatives, $p = \frac{\partial Z}{\partial x}$ and $q = \frac{\partial Z}{\partial y}$. By knowing the relative orientation of the surface, light source, and viewer, we can compute a reflectance map, which projects surface gradients onto expected image pixel intensities. Most SFS techniques utilize such a reflectance map, but a reflectance map only enables an algorithm to directly calculate possible surface orientations; consequently, various SFS approaches differ in large part by how they choose to combine local orientation information to reconstruct a 3D shape [3].

In general, SFS approaches can be grouped into four categories: propagation approaches, local approaches, linear approaches, and minimization-based approaches [3]. Propagation approaches, such as that described by Horn, were some of the first to be researched, and rely on propagating shape information outward from a set of initially chosen singular points at which the depth is known; local approaches compute the depth at each point based on the point’s value, and the first and the second derivatives; linear approaches rely on a linear approximation of the reflectance function; and minimization approaches, explored by many different researchers, recover a 3D shape by minimizing an energy function [3].

Unfortunately, the approaches listed above are generally intolerant to noise or rely on assumptions that are hard to be re-
alized for rooftop images. For instance, in propagation-based methods, noise can accumulate as it propagates from one part of the image to the other. Moreover, most of these approaches are unable to handle non-Lambertian reflectance, interreflections, and other complex lighting. Furthermore, reconstructing shape from shading without constraints is an underconstrained problem, so for any given image, multiple recovered shapes are possible [3].

Consequently, some practical approaches rely on user input during segmentation to achieve satisfactory results [5]. Another way to overcome these limitations is by utilizing constraints based on the type of objects being reconstructed [6].

In this paper, we propose a minimization technique that does impose strong constraints on the recovered 3D shape. We are able to impose these constraints because we know that the objects that we are recovering are rooftops, which share common characteristics such as straight lines, flat, downwardly-pitched surfaces, and typically are made of asphalt tiled roofing material.

2. RECONSTRUCTING 3D ROOFTOPS

The algorithm consists of four general steps: extracting each rooftop from a larger image; preprocessing to remove details like vents; segmenting into polygonal planar surfaces; and generating a 3D model from the planar surfaces.

2.1. Extracting Roofs From Aerial Images

The first step is performed using a hierarchical feature-based image segmentation algorithm [2]. The input of this step is a satellite or aerial image of an area containing buildings with pitched roofs, and the output of this step is a set of regions corresponding to the rooftops in the input image. These regions are then cropped automatically out of the image, and for each individual rooftop image, we perform the remaining steps of the algorithm.

2.2. Removing Insignificant Rooftop Details

Removing outlier regions involves performing a threshold operation to find very bright or very dark small regions within the rooftop boundary. These regions typically correspond to rooftop vents. In a histogram-equalized image, we found that pixels with intensities greater than 180 on a 0 to 255 scale typically corresponded with white plastic vents, and pixels with intensities less than 90 typically corresponded to black plastic vents.

Once these outlier regions have been identified, extremely small regions are removed by a morphological opening operation. The remaining regions are then used as a mask for inpainting [7], thus erasing the vents and using surrounding pixels to fill in the area that they previously occupied. Figure 2 illustrates the output of this vent removal process.

Fig. 2: An example of an original rooftop image (left) and the processed image with the vents removed (right)

Next, we apply an edge preserving filter to reduce the rough texture of the roof. The classic median filter produces good results, but we found that mean-shift filtering [8] produced images that best captured the relevant edges while smoothing irrelevant details.

Finally, we apply a histogram equalization to the image to normalize the contrast within the image. This step is important for the following steps, which rely on the contrast between roof regions to identify edges.

2.3. Identifying Planar Rooftop Surfaces

Segmentation starts by identifying straight edges in the image. We used a Canny edge detector with a lower threshold of 250 and an upper threshold of 300 [9]. Next, we applied a morphological dilation operation to the binary edge image to connect nearby edges and identify large edgeless regions. These regions are then used as markers for a watershed segmentation [10] of the rooftop image, with the strong edges that were previously identified used as barriers.

Finally, for each segment found using the watershed algorithm, we apply the Douglas Peucker algorithm [11] to approximate the contour with a polygon of just a few vertices. This usually results in polygonal regions with three to five
vertices. Figure 3 illustrates the results of the edge-detection, segmentation, and shape simplification steps.

![Figure 3](image)

**Fig. 3**: (a) The preprocessed image, (b) the image after edge detection, (c) the image after watershed segmentation, and (d) the image after simplifying the regions into simpler polygons.

### 2.4. Deducing Depth From Shading

To estimate the depth from shading information, we need to first know the Bidirectional Reflectance Distribution Function (BRDF) that best models the rooftop material. By default, OpenGL uses a shading model called Blinn-Phong, which models a generic BRDF as the weighted sum of three components: an ambient component, a diffuse component, and a glossy specular component [12]. The Blinn-Phong model does not model the roughness of roof shingles well, however. The roughness tends to reduce the dependence of the perceived brightness of a surface on the viewing angle. A better model for rough surfaces is the Oren-Nayer model [13]. Here the reflected radiance $L_r$ is a function of the incident and reflected angles, as well as two parameters, $\rho$, the albedo of the surface, and $\sigma$, the roughness of the surface. Roofing shingles are well modeled with $\rho = 0.20$ and $\sigma = 0.82$ [14].

We extract the lighting information from the acquisition geometry included along with the image metadata. This acquisition geometry describes the sensor and the sun azimuth and elevation angles [2].

Once we have BRDF and lighting information, we can render our hypothesized 3D shapes using OpenGL. The rendered image is then compared against the original image (with vents removed) to calculate a fitness measure of the hypothesized 3D shape. In the algorithm, a simple correlation with the original image is utilized as a fitness measure, $F$, where

$$F = \sum_{(x, y) \in I_1} (I_1(x, y) - I_2(x, y))^2.$$

$I_1$ and $I_2$ are the original and rendered images, respectively. A lower value for $F$ indicates a better model.

The algorithm then perturbs the $z$ coordinates of the vertices in the rendered rooftop model slightly and renders it again to reduce $F$. The $z$ coordinates are perturbed according to a gradient descent equation. The set of all $z$ values for all of the vertices are used to form a vector $\mathbf{z}$. Thus, given a set of $z$ values $\mathbf{z}_i$ at the $i$th iteration, the set of $z$ values at the $(i + 1)$th iteration is computed by:

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \nabla F(\mathbf{z}_i).$$

Once a reconstruction is established that achieves a desired fitness measure, the algorithm stops iterating and the final image is rendered. Two examples of reconstructed 3D rooftops using this processes are shown in Figure 4. The results on the second row of this figure correspond to the example shown in Figure 4.

![Figure 4](image)

**Fig. 4**: Examples of the rendered generated 3D models.

### 3. RESULTS

In this section, we present some results of the proposed algorithm along with issues related to the implementation.

#### 3.1. Accuracy of Roof Pitch Estimation

The accuracy of the roof pitch estimation was tested on several buildings. Our main constraint here was the ground truth had to be created manually by going to the site and physically measuring rooftop slopes. We found that the estimated roof pitch was usually within a few degrees of the true roof pitch. Table 1 lists the true and estimated pitches for some of the tested buildings.

<table>
<thead>
<tr>
<th>Rooftop Image</th>
<th>Actual Pitch</th>
<th>Estimated Pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 (Chilcotin House)</td>
<td>26°</td>
<td>28°</td>
</tr>
<tr>
<td>#2 (Kelowna Res.)</td>
<td>26°</td>
<td>28°</td>
</tr>
<tr>
<td>#3 (Curtis Road #1)</td>
<td>29°</td>
<td>30°</td>
</tr>
<tr>
<td>#4 (Curtis Road #2)</td>
<td>30°</td>
<td>30°</td>
</tr>
<tr>
<td>#5 (House on 194A St)</td>
<td>20°</td>
<td>23°</td>
</tr>
<tr>
<td>#6 (Cross Twn Food Mkt)</td>
<td>26°</td>
<td>28°</td>
</tr>
<tr>
<td>#7 (Fort Lang. Hall)</td>
<td>28°</td>
<td>28°</td>
</tr>
<tr>
<td>#8 (Aldergrove Store)</td>
<td>47°</td>
<td>45°</td>
</tr>
<tr>
<td>#9 (Langley Hall)</td>
<td>44°</td>
<td>45°</td>
</tr>
<tr>
<td>#10 (C. &amp; G. Howe Schl)</td>
<td>16°</td>
<td>18°</td>
</tr>
</tbody>
</table>

| | Avg. Error: | 1.5° |

**Table 1**: Rooftop pitch estimation.
3.2. Processor and Memory Utilization

The implementation of this algorithm operates in a reasonable time and with reasonable memory requirements. Our test machine was a Toshiba Satellite laptop with an AMD Athlon(tm) 64 X2 processor running at 800 Mhz. Our implementation was not multi-threaded, so it utilized only one of the two cores available on the machine. For the 3D reconstruction of a rooftop of size $160 \times 160$ pixels, the algorithm takes about 5.8 seconds to complete.

3.3. Limitations

This algorithm requires sufficient resolution and contrast to identify rooftop features. We found that the algorithm works well with images that have a minimum resolution of about 0.2 meters per pixel. At that resolution, edges that are shorter than approximately 1 m are sometimes lost, because such short edges are ignored as potential noise or combined with other nearby edges. Additionally, those images taken on clear days in the morning and evening typically have the highest contrast, and therefore result in more robust segmentation.

Additionally, the proposed algorithm currently works reliably only for relatively simple rooftop shapes, such as those with two to five planar regions that are connected edge-to-edge. Ledges and sharp discontinuities in the roof surface cannot easily be accounted for, because, in general, two disconnected rooftop surfaces that are separated only by a vertical gap look exactly the same as two connected rooftop surfaces when seen from above. Therefore, the algorithm assumes that the roof is a continuous surface, without shear vertical drops or ledges.

4. CONCLUSIONS

We have presented an algorithm for estimating the three-dimensional shape of a pitched roof, using just a single image. This approach differs from previous approaches, many of which require specialized equipment for laser ranging, or multiple images taken from different vantage points. We found that this algorithm usually accurately estimates the 3D shape of gabled rooftops, especially when the images have high resolution and good contrast.

Our objective is to create the 3D modeled rooftops of buildings given the top view of a rooftop. This work adds on to the previous work in our group that addressed the height estimation of buildings (average height from the borders of the roof lines to ground). Using the rooftop models, a more complete reconstruction using singular electro-optical imagery is possible.

5. REFERENCES